

**PATENT APPLICATION
DOCKET NO. 200300161-1**

**IN THE
UNITED STATES PATENT AND TRADEMARK OFFICE**

INVENTOR(S): John L. Manuel

CONFIRMATION NO: 7814

SERIAL NO.: 10/702,151

GROUP ART UNIT: 2194

FILED: November 5, 2003

EXAMINER: Seye, Abdou K

**SUBJECT: SYSTEM AND METHOD FOR CREATING
A BEST-MATCH OBJECT AT RUN TIME**

U.S. PATENT AND TRADEMARK OFFICE
COMMISSIONER OF PATENTS
ALEXANDRIA, VA 22313

APPELLANTS'/APPLICANTS' OPENING BRIEF ON APPEAL

1. REAL PARTY IN INTEREST.

The real party in interest is Hewlett-Packard Development Company, LP, a limited partnership established under the laws of the State of Texas and having a principal place of business at 11445 Compaq Center Drive W. Houston TX 77070 USA (hereinafter "HPDC"). HPDC is a Texas limited partnership and is a wholly-owned affiliate of Hewlett-Packard Company, a Delaware Corporation, headquartered in Palo Alto, CA. The general or managing partner of HPDC is HPQ Holding, LLC.

2. RELATED APPEALS AND INTERFERENCES.

There are no other appeals or interferences known to Appellants, Appellants' legal representative or the Assignee which will affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

3. STATUS OF CLAIMS.

Claims 1-23, 31, and 33-44 are pending and stand rejected. Claims 24-30 and 32 have been cancelled. All rejections are appealed.

4. STATUS OF AMENDMENTS.

Claims 24-30 were cancelled following entry of the final rejections. All other previous amendments have been entered.

5. SUMMARY OF CLAIMED SUBJECT MATTER.

Claim 1 recites a method for creating a best-match object at run time. The method includes receiving a request for an object. See, e.g., Specification, paragraph [0043], page 12, lines 10-20 and Figure 4, step 402. Object proxies are polled for a confidence level representing the capability of each respective proxy to generate the requested object. See, e.g., Specification, paragraph [0043], page 12, lines 10-20 and Figure 4, step 404. One of the proxies is selected based on the polled confidence level. See, e.g., Specification, paragraph [0043], page 12, lines 10-20 and Figure 4, step

406The selected proxy is directed to create the object. See, e.g., Specification, paragraph [0043], page 12, lines 10-20 and Figure 4, step 408

Claim 8 recites a system that includes an object generator and a processor operable to execute the object generator. See, e.g., Specification, paragraphs [0019]-[0020], page 4, line 21 through page 5, line 6; paragraphs [0022]-[0023], page 5, line 15 through page 6, line 5; Figure 1, elements 110 and 200. The object generator includes instructions that when executed by the processor function as (a) means for receiving indicia of an object to be created; (b) means for identifying a select one of a plurality of object proxies responsive to a respective confidence level associated with each object proxy; and (c) means for directing the selected object proxy to create the object. See, e.g., Specification, paragraphs [0019]-[0020], page 4, line 21 through page 5, line 6; paragraphs [0022]-[0023], page 5, line 15 through page 6, line 5; Figure 1, elements 110, 200, 210, 220, 230, and 240.

Claim 14 recites a system that includes an object generator and a processor operable to execute the object generator. See, e.g., Specification, paragraphs [0019]-[0020], page 14, line 21 through page 5, line 6; paragraphs [0022]-[0023], page 5, line 15 through page 6, line 5; Figure 1, elements 110 and 200. The object generator includes instructions that when executed by the processor function as (a) an object factory configured to poll object proxies capable of producing respective objects responsive to system needs; and (b) a pool including the object proxies for producing the object. See, e.g., Specification, paragraphs [0019]-[0020], page 4, line 21 through page 5, line 6; paragraphs [0022]-[0023], page 5, line 15 through page 6, line 5; Figure 1, elements 110 and 200. The pool is configured to receive indicia of the object from the object factory, and each of the plurality of object proxies is configured to return a respective confidence level responsive to the indicia. See, e.g., Specification, paragraphs [0030]-[0033], page 7, line 25 through page 9, line 2; Figure 1, elements 110 and 200, 220, and 230.

Claim 30 recites a system that includes an object generator and a processor operable to execute the object generator. See, e.g., Specification, paragraphs [0019]-[0020], page 14, line 21 through page 5, line 6; paragraphs [0022]-[0023], page 5, line 15 through page 6, line 5; Figure 1, elements 110 and 200. The object generator includes instructions that when executed by the processor function as (a) an object factory configured to receive a device identifier; (b) a pool having an interface configured to communicate with the object factory, the pool containing object proxies capable of producing respective objects; and (c) an object store coupled to the pool and configured to receive and retain objects generated by selected object proxies. See, e.g., Specification, paragraphs [0030]-[0033], page 7, line 25 through page 9, line 2; Figure 1, elements 110 and 200, 220, and 230. The object factory is configured to poll a plurality of object proxies for a confidence level representing the capability of the respective object proxy to generate an object suited for operating with a device responsive to the device identifier. See, e.g., Specification, paragraphs [0030]-[0033], page 7, line 25 through page 9, line 2; Figure 1, elements 110 and 200, 220, and 230.

Claim 33 recites a method for creating a best-match object at run time. The method includes loading a set of object proxies. See, e.g., Specification, paragraph [0044], page 12, line 21 through page 13, line 3; Figure 5, step 502. Indicia of a desired object for communicating with a peripheral device is received. See, e.g., Specification, paragraph [0044], page 12, line 21 through page 13, line 3; Figure 5, step 504. Each of the object proxies is directed to forward a confidence level representing the capability of each respective proxy to generate the desired object responsive to the indicia. See, e.g., Specification, paragraph [0045], page 12, line 27 through page 13, line 3; Figure 5, step 508. A confidence level associated with an object proxy is received. See, e.g., Specification, paragraph [0045], page 12, line 27 through page 13, line 3; Figure 5, step 510. The confidence level is compared to a maximum confidence level. See, e.g., Specification, paragraph [0045], page 12, line 27 through page 13, line 3; Figure 5, step 512. When the confidence level matches the maximum confidence level, the associated object proxy is directed to generate an object. See, e.g., Specification, paragraph [0047], page 13, lines 21-24; Figure 5, step 524. Otherwise the confidence

level is recorded. See, e.g., Specification, paragraph [0045], page 12, line 27 through page 13, line 3; Figure 5, step 514. It is determined if the confidence level exceeds the confidence level associated with a previously recorded confidence level. See, e.g., Specification, paragraph [0046], page 13, lines 4-20; Figure 5, step 516. When the confidence level exceeds a previously recorded confidence level, an object proxy identifier is recorded. See, e.g., Specification, paragraph [0046], page 13, lines 4-20; Figure 5, step 518. Otherwise, it is determined if there are additional object proxies in the set. See, e.g., Specification, paragraph [0046], page 13, lines 4-20; Figure 5, step 520. When there are additional object proxies, the receiving a confidence level, comparing, and determining if the confidence level exceeds steps are repeated. See, e.g., Specification, paragraphs [0044]-[0047], page 12, lines 21 through page 13, line 24. Otherwise, the object proxy identifier is used to direct the associated object proxy to generate an object. See, e.g., Specification, paragraph [0047], page 13, lines 21-24; Figure 5, step 524.

Claim 34 recites a computer-readable medium that includes various pieces of logic configured to implement various functions. See, e.g., Specification, paragraph [0048], page 13, lines 25-32. The medium stores logic configured to load a set of object proxies, each object proxy configured to generate a respective object. See, e.g., Specification, paragraph [0044], page 12, line 21 through page 13, line 3; Figure 5, step 502. The medium stores logic configured to receive indicia of a desired object for communicating with a peripheral device. See, e.g., Specification, paragraph [0044], page 12, line 21 through page 13, line 3; Figure 5, step 504. The medium stores logic configured to direct each of the object proxies to forward a confidence level representing the capability of each respective proxy to generate the desired object. See, e.g., Specification, paragraph [0045], page 12, line 27 through page 13, line 3; Figure 5, step 508. Included is logic configured to receive the confidence level from respective object proxies. See, e.g., Specification, paragraph [0045], page 12, line 27 through page 13, line 3; Figure 5, step 510. The medium also includes logic configured to compare the confidence level to a maximum confidence level. See, e.g., Specification, paragraph [0045], page 12, line 27 through page 13, line 3; Figure 5, step

512. When the confidence level matches the maximum confidence level, the associated object proxy is directed to generate an object. See, e.g., Specification, paragraph [0047], page 13, lines 21-24; Figure 5, step 524. Otherwise, the logic records the confidence level; and determines if the confidence level exceeds the confidence level associated with a previously recorded confidence level. See, e.g., Specification, paragraph [0045], page 12, line 27 through page 13, line 3; Figure 5, steps 514 and 516. When the confidence level exceeds a previously recorded confidence level, the logic records an object proxy identifier. See, e.g., Specification, paragraph [0046], page 13, lines 4-20; Figure 5, step 518. Otherwise, the logic determines if there are additional object proxies in the set. See, e.g., Specification, paragraph [0046], page 13, lines 4-20; Figure 5, step 520. When there are additional object proxies, the logic receives a confidence level associated with an object proxy that has not reported a confidence level, and repeats the maximum confidence level and previously recorded confidence level comparisons. See, e.g., Specification, paragraphs [0044]-[0047], page 12, lines 21 through page 13, line 24. Otherwise, the logic uses the object proxy identifier to direct the associated object proxy to generate an object. See, e.g., Specification, paragraph [0047], page 13, lines 21-24; Figure 5, step 524.

Claim 35 recites a method for creating a best-match printer driver. The method includes receiving a request to use a printer. See, e.g., Specification, paragraph [0026]-[0027], page 6, lines 20 through page 7, line 9. Printer driver proxies are polled for a confidence level representing the capability of each respective printer driver proxy to generate a driver that when applied to data and forwarded to the printer will produce a useful representation of the data. See, e.g., Specification, paragraph [0030], page 7, line 25 through page 8, line 6; paragraph [0050], page 14, lines 16-32. One of the printer driver proxies is selected based on the polled confidence level. See, e.g., Specification, paragraph [0051], page 14, line 30 through page 15, line 16. The selected printer driver proxy is directed to generate the driver. See, e.g., Specification, paragraph [0051], page 14, line 30 through page 15, line 16.

Claim 40 recites a computer-readable medium. See, e.g., Specification, paragraph [0048], page 13, lines 25-32. That medium stores logic for implementing various functions. See, e.g., Specification, paragraph [0048], page 13, lines 25-32. The medium stores logic configured to receive a request to use a printer. See, e.g., Specification, paragraph [0026]-[0027], page 6, lines 20 through page 7, line 9. The medium stores logic configured to poll printer driver proxies for a confidence level representing the capability of each respective printer driver proxy to generate a driver that when applied to data and forwarded to the printer will produce a useful representation of the data. See, e.g., Specification, paragraph [0030], page 7, line 25 through page 8, line 6; paragraph [0050], page 14, lines 16-32. Included is logic configured to select one of the printer driver proxies based on the polled confidence level. See, e.g., Specification, paragraph [0051], page 14, line 30 through page 15, line 16. The medium also includes logic configured to direct the selected printer driver proxy to generate the driver. See, e.g., Specification, paragraph [0051], page 14, line 30 through page 15, line 16.

6. GROUNDS FOR REJECTION TO BE REVIEWED.

- A. Claims 24-30, 34, and 40-44 stand rejected under 35 USC §101 as being directed to non-statutory subject matter.
- B. Claims 1-31 and 33-44 stand rejected under 35 USC §103 as being as unpatentable over USPN 7,181,382 issued to Shier in view of in view of USPN 6,477,520 issued to Malaviya.

7. ARGUMENT.

Grounds For Rejection A – Claims 24-30, 34, and 40-44 stand rejected under 35 USC §101 as being directed to non-statutory subject matter.

CLAIM REJECTIONS – 35 USC §101: The Examiner rejected Claims 24-30, 34, and 40-44 as being directed to non-statutory subject matter. Claims 24-30 have been

cancelled. Paragraph [0031] has been amended to address some of the Examiner's concerns. In particular, "computer readable media" is no longer defined to include paper with printed program instructions.

The Examiner, however, also stated:

Claims 24-30, 34 40-44 are rejected under 35 U.S.C. 101 because the claimed invention is directed to a computer product comprising: logic configured to receive; logic configured to poll; logic configured to select and logic configured to direct. While not explicitly describing whether these logic are instructions stored on a computer readable medium and executed; therefore the computer product appears to be comprised of software without claiming execution of these logic.

The Applicant respectfully disagree with the Examiner's assessment. Claims 34 and 40-44 each recite a computer readable medium comprising logic configured to implement a number of specified functions. Paragraph [0048] of the specification states: "Any process descriptions or functions in the flow charts presented in FIGs. 4 and 5 should be understood to represent modules, segments, or portions of code or logic, which include one or more executable instructions for implementing specific logical functions in the associated process." In light of this passage, the term "logic," as used in the Claims, is defined as computer executable instructions for implementing specified logical functions."

Because the recited computer readable medium comprises the recited logic and the logic is defined as computer executable instructions, that computer readable medium stores those computer executable instructions. Thus, Claims 34 and 40-44 are not software per se but recite a medium storing computer executable instructions (logic) that implement specified logical when executed.

For at least these reasons, Claim 34 and Claims 40-44 are directed to statutory subject matter under §101.

Grounds For Rejection B – Claims 1-31 and 33-44 stand rejected under 35 USC §103 as being as unpatentable over USPN 7,181,382 issued to Shier in view of in view of USPN 6,477,520 issued to Malaviya.

Claim 1 is directed to a method for creating a best-match object at run time and recites the following:

1. receiving a request for an object;
2. polling object proxies for a confidence level representing the capability of each respective proxy to generate the requested object;
3. selecting one of the proxies based on the polled confidence level; and
4. directing the selected proxy to create the object.

The Examiner asserts that Shier, col. 15, lines 7-58 along with Figs 3A, 3B and 4 teach polling object proxies for a confidence level representing the capability of each respective proxy to generate the requested object. The Applicant respectfully disagrees.

As explained in paragraph [0018] and [0030], [0036], and [0043] of the Specification, each object proxy is capable of generating a particular object and each encapsulates information that describes what that particular object it is configured to create. When polled as to whether or not it can generate a requested object, each object proxy returns a confidence level. A given one of the object proxies is selected based on its response to the poll. The selected object proxy then generates the requested object.

The passage relied upon by the Examiner mentions nothing of polling object proxies – that is – polling a plurality of object proxies for any reason let alone for a confidence level that each can generate a requested object. The passage relied upon by the Examiner describes a test application that generates of an object that facilitates the communication of instruction produced by the test application. Such an instruction may be an instruction that a particular key board key be actuated. Shier, col. 15, lines

4-16. Shier's test application is not an object proxy that is polled for a confidence level. The cited passage also describes a user mode framework that generates a proxy object. Shier, col. 15, lines 17-26. The user mode framework is not an object proxy that is polled for a confidence level. The remainder of the cited passage discusses communications with Shier's proxy object. Shier, col. 15, lines 27-58. While the test application and the user framework generate objects, those two components are not polled for a confidence level.

Consequently, Shier fails to teach or suggest polling object proxies for a confidence level representing the capability of each respective proxy to generate the requested object. Malaviya is silent on this matter. For at least these reasons, Claim 1 is patentable over the cited references as are Claims 2-7 which depend from Claim 1

Claim 8 recites a system that includes an object generator and a processor operable to execute the object generator. The object generator includes instructions that when executed by the processor function as:

1. means for receiving indicia of an object to be created;
2. means for identifying a select one of a plurality of object proxies responsive to a respective confidence level associated with each object proxy; and
3. means for directing the selected object proxy to create the object.

As discussed above Shier and Malaviya fail to teach or suggest polling object proxies for a confidence level representing the capability of each respective proxy to generate the requested object. For the same reasons, those references also fail to teach or suggest means for identifying a select one of a plurality of object proxies responsive to a respective confidence level associated with each object proxy.

For at least these reasons, Claim 8 and Claims 9-13 which depend from Claim 8 are patentable over Shier and Malaviya.

Claim 14 recites a system that includes an object generator and a processor operable to execute the object generator. The object generator includes instructions that when executed by the processor function as:

1. an object factory configured to poll object proxies capable of producing respective objects responsive to system needs; and
2. a pool including the object proxies for producing the object, the pool configured to receive indicia of the object from the object factory and each of the plurality of object proxies configured to return a respective confidence level responsive to the indicia.

As discussed above Shier and Malaviya fail to teach or suggest polling object proxies for a confidence level representing the capability of each respective proxy to generate the requested object. For the same reasons, those references also fail to teach or suggest “a pool including the object proxies . . . each of the plurality of object proxies configured to return a respective confidence level responsive to the indicia.”

For at least these reasons, Claim 14 and Claims 12-23 which depend from Claim 14 are patentable over Shier and Malaviya.

Claim 31 recites a system that includes an object generator and a processor operable to execute the object generator. The object generator includes instructions that when executed by the processor function as:

1. an object factory configured to receive a device identifier;
2. a pool having an interface configured to communicate with the object factory, the pool containing object proxies capable of producing respective objects; and

3. an object store coupled to the pool and configured to receive and retain objects generated by selected object proxies;
4. wherein the object factory is configured to poll a plurality of object proxies for a confidence level representing the capability of the respective object proxy to generate an object suited for operating with a device responsive to the device identifier.

As discussed above Shier and Malaviya fail to teach or suggest polling object proxies for a confidence level representing the capability of each respective proxy to generate the requested object. For the same reasons, those references also fail to teach or suggest an object factory that is “configured to poll a plurality of object proxies for a confidence level representing the capability of the respective object proxy to generate an object suited for operating with a device responsive to the device identifier.”

For at least these reasons, Claim 31 is patentable over Shier and Malaviya.

Claim 33 is directed to a method for creating a best-match object at run time and recites the following:

1. loading a set of object proxies;
2. receiving indicia of a desired object for communicating with a peripheral device;
3. directing each of the object proxies to forward a confidence level representing the capability of each respective proxy to generate the desired object responsive to the indicia;
4. receiving a confidence level associated with an object proxy;
5. comparing the confidence level to a maximum confidence level, when the confidence level matches the maximum confidence level, directing the associated object proxy to generate an object, otherwise, recording the confidence level; and

6. determining if the confidence level exceeds the confidence level associated with a previously recorded confidence level, when the confidence level exceeds a previously recorded confidence level, recording an object proxy identifier, otherwise, determining if there are additional object proxies in the set, when there are additional object proxies, repeating the receiving a confidence level, comparing, and determining if the confidence level exceeds steps, otherwise, using the object proxy identifier to direct the associated object proxy to generate an object.

As discussed above Shier and Malaviya fail to teach or suggest polling object proxies for a confidence level representing the capability of each respective proxy to generate the requested object. For the same reasons, those references also fail to teach or suggest a method that includes (a) directing each of the object proxies to forward a confidence level representing the capability of each respective proxy to generate the desired object responsive to the indicia and (b) receiving a confidence level associated with an object proxy.

For at least these reasons, Claim 33 is patentable over Shier and Malaviya.

Claim 34 is directed to a computer readable medium that includes logic configured to implement the method of Claim 33. For at least the same reasons Claim 33 is patentable, so is Claim 34.

Claim 35 is directed to a method for creating a best-match printer driver and recites the following:

1. receiving a request to use a printer;

2. polling printer driver proxies for a confidence level representing the capability of each respective printer driver proxy to generate a driver that when applied to data and forwarded to the printer will produce a useful representation of the data;
3. selecting one of the printer driver proxies based on the polled confidence level; and
4. directing the selected printer driver proxy to generate the driver.

As discussed above Shier and Malaviya fail to teach or suggest polling object proxies for a confidence level representing the capability of each respective proxy to generate the requested object. For the same reasons, those references also fail to teach or suggest a method that includes “polling printer driver proxies for a confidence level representing the capability of each respective printer driver proxy to generate a driver that when applied to data and forwarded to the printer will produce a useful representation of the data.”

Furthermore, addressing Claim 35, the Examiner does not assert that Shier or Malaviya mentions anything related to printer drivers let alone directing the selected printer driver proxy to generate the driver or polling printer driver proxies for information of any kind.

For at least these reasons, Claim 35 and Claims 36-39 which depend from Claim 34 are patentable over Shier and Malaviya.

Claim 40 is directed to a computer readable medium that includes logic configured to implement the method of Claim 35. For at least the same reasons Claim 35 is patentable, so is Claim 40 and Claims 41-44 which depend from Claim 40..

Conclusion: In view of the foregoing remarks, the Applicant respectfully submits that the pending claims are in condition for allowance. Consequently, early and favorable action allowing these claims and passing the application to issue is earnestly solicited.

Respectfully submitted,
John L. Manuel

By /Jack H. McKinney/
Jack H. McKinney
Reg. No. 45,685

December 8, 2008

APPENDIX OF CLAIMS INVOLVED IN THE APPEAL

1. (original) A method for creating a best-match object at run time, comprising the steps of:

receiving a request for an object;
polling object proxies for a confidence level representing the capability of each respective proxy to generate the requested object;
selecting one of the proxies based on the polled confidence level; and
directing the selected proxy to create the object.

2. (original) The method of claim 1, wherein the step of receiving a request for an object comprises receiving indicia of a peripheral device.

3. (original) The method of claim 2, wherein indicia comprises a device identifier.

4. (original) The method of claim 1, wherein the step of selecting one of the proxies comprises comparing each confidence level with a previously received confidence level.

5. (original) The method of claim 1, wherein the step of selecting one of the proxies comprises storing an index associated with a proxy having a greater confidence level.

6. (original) The method of claim 1, wherein the step of directing the select one of the proxies to create the object generates a peripheral device driver.

7. (original) The method of claim 1, further comprising the step of:
registering a new proxy capable of creating an object designated for use with a new peripheral device.

8. (previously presented) A system, comprising an object generator and a processor operable to execute the object generator, the object generator including instructions that when executed by the processor function as:

means for receiving indicia of an object to be created;

means for identifying a select one of a plurality of object proxies responsive to a respective confidence level associated with each object proxy; and

means for directing the selected object proxy to create the object.

9. (previously presented) The system of claim 8, wherein the means for receiving is responsive to a user interface.

10. (previously presented) The system of claim 8, wherein the means for receiving is responsive to a communication from a device associated with the object.

11. (previously presented) The system of claim 8, wherein the means for identifying a select one of a plurality of object proxies comprises means for comparing each respective confidence level with a previously received confidence level.

12. (previously presented) The system of claim 8, wherein the means for identifying a select one of a plurality of object proxies comprises means for comparing each confidence level with a maximum confidence level.

13. (previously presented) The system of claim 12, wherein the means for identifying a select one of a plurality of object proxies identifies an object proxy that returns the maximum confidence level as the selected object proxy.

14. (previously presented) A system, comprising an object generator and a processor operable to execute the object generator, the object generator including instructions that when executed by the processor function as:

an object factory configured to poll object proxies capable of producing respective objects responsive to system needs; and

a pool including the object proxies for producing the object, the pool configured to receive indicia of the object from the object factory and each of the plurality of object proxies configured to return a respective confidence level responsive to the indicia.

15. (previously presented) The system of claim 14, further comprising:
an interface associated with the object factory, the interface configured to receive a request for the object.

16. (previously presented) The system of claim 15, wherein the interface is configured to communicate with a user interface.

17. (previously presented) The system of claim 15, wherein the interface is configured to communicate with a device that will interact with the object.

18. (previously presented) The system of claim 15, wherein the interface is configured to receive a device identifier.

19. (previously presented) The system of claim 15, wherein the interface is configured to receive a device identifier associated with a printer.

20. (previously presented) The system of claim 14, wherein the object factory comprises a comparator configured to determine which of a first confidence level associated with a first object proxy and a second confidence level associated with a second object proxy is more likely to produce an object most responsive to the system need.

21. (previously presented) The system of claim 20, wherein when the comparator is configured to recognize a maximum confidence level, the object factory is configured to direct the object proxy associated with the maximum confidence level to create an object.

22. (previously presented) The system of claim 20, wherein when the comparator fails to recognize a maximum confidence level, the object factory is configured to direct the object proxy associated with the greatest confidence level to create an object.

23. (previously presented) The system of claim 14, further comprising:
an object store configured to receive an object generated by an object proxy.

24-30. (cancelled)

31. (previously presented) A system, comprising an object generator and a processor operable to execute the object generator, the object generator including instructions that when executed by the processor function as:

- an object factory configured to receive a device identifier;
- a pool having an interface configured to communicate with the object factory, the pool containing object proxies capable of producing respective objects; and
- an object store coupled to the pool and configured to receive and retain objects generated by selected object proxies;

wherein the object factory is configured to poll a plurality of object proxies for a confidence level representing the capability of the respective object proxy to generate an object suited for operating with a device responsive to the device identifier.

32. (cancelled)

33. (original) A method for creating a best-match object at run time, comprising the steps of:

- loading a set of object proxies;
- receiving indicia of a desired object for communicating with a peripheral device;
- directing each of the object proxies to forward a confidence level representing the capability of each respective proxy to generate the desired object responsive to the indicia;
- receiving a confidence level associated with an object proxy;

comparing the confidence level to a maximum confidence level, when the confidence level matches the maximum confidence level, directing the associated object proxy to generate an object, otherwise, recording the confidence level; and

determining if the confidence level exceeds the confidence level associated with a previously recorded confidence level, when the confidence level exceeds a previously recorded confidence level, recording an object proxy identifier, otherwise, determining if there are additional object proxies in the set, when there are additional object proxies, repeating the receiving a confidence level, comparing, and determining if the confidence level exceeds steps, otherwise, using the object proxy identifier to direct the associated object proxy to generate an object.

34. (original) A computer-readable medium, comprising:

logic configured to load a set of object proxies, each object proxy configured to generate a respective object;

logic configured to receive indicia of a desired object for communicating with a peripheral device;

logic configured to direct each of the object proxies to forward a confidence level representing the capability of each respective proxy to generate the desired object;

logic configured to receive the confidence level from respective object proxies;

logic configured to compare the confidence level to a maximum confidence level, when the confidence level matches the maximum confidence level, the associated object proxy is directed to generate an object, otherwise, the logic records the confidence level; and determines if the confidence level exceeds the confidence level associated with a previously recorded confidence level, when the confidence level exceeds a previously recorded confidence level, the logic records an object proxy identifier, otherwise, the logic determines if there are additional object proxies in the set, when there are additional object proxies, the logic receives a confidence level associated with an object proxy that has not reported a confidence level, and repeats the maximum confidence level and previously recorded confidence level comparisons, otherwise, the logic uses the object proxy identifier to direct the associated object proxy to generate an object.

35. (original) A method for creating a best-match printer driver, comprising the steps of:

receiving a request to use a printer;

polling printer driver proxies for a confidence level representing the capability of each respective printer driver proxy to generate a driver that when applied to data and forwarded to the printer will produce a useful representation of the data;

selecting one of the printer driver proxies based on the polled confidence level;
and

directing the selected printer driver proxy to generate the driver.

36. (original) The method of claim 35, wherein the step of receiving a request to use a printer comprises receiving a device identifier.

37. (original) The method of claim 35, wherein the step of receiving a request to use a printer comprises receiving indicia of a printer capability.

38. (original) The method of claim 35, wherein the step of selecting one of the printer driver proxies comprises comparing each confidence level with a previously received confidence level.

39. (original) The method of claim 35, wherein the step of selecting one of the printer driver proxies comprises storing an index associated with a printer driver proxy having a greater confidence level.

40. (original) A computer-readable medium, comprising:

logic configured to receive a request to use a printer;

logic configured to poll printer driver proxies for a confidence level representing the capability of each respective printer driver proxy to generate a driver that when applied to data and forwarded to the printer will produce a useful representation of the data;

logic configured to select one of the printer driver proxies based on the polled confidence level; and

logic configured to direct the selected printer driver proxy to generate the driver.

41. (original) The computer-readable medium of claim 40, wherein the logic configured to receive a request to use a printer is configured to receive a device identifier.

42. (original) The computer-readable medium of claim 40, wherein the logic configured to receive a request to use a printer is configured to receive indicia of a printer capability.

43. (original) The computer-readable medium of claim 40, wherein the logic configured to select one of the printer driver proxies is configured to compare confidence levels with a previously received confidence level.

44. (original) The computer-readable medium of claim 40, wherein the logic configured to select one of the printer driver proxies is configured to store an index associated with a printer driver proxy having a greater confidence level.

Evidence Appendix

There is no extrinsic evidence to be considered in this Appeal. Therefore, no evidence is presented in this Appendix.

Related Proceedings Appendix

There are no related proceedings to be considered in this Appeal. Therefore, no such proceedings are identified in this Appendix.